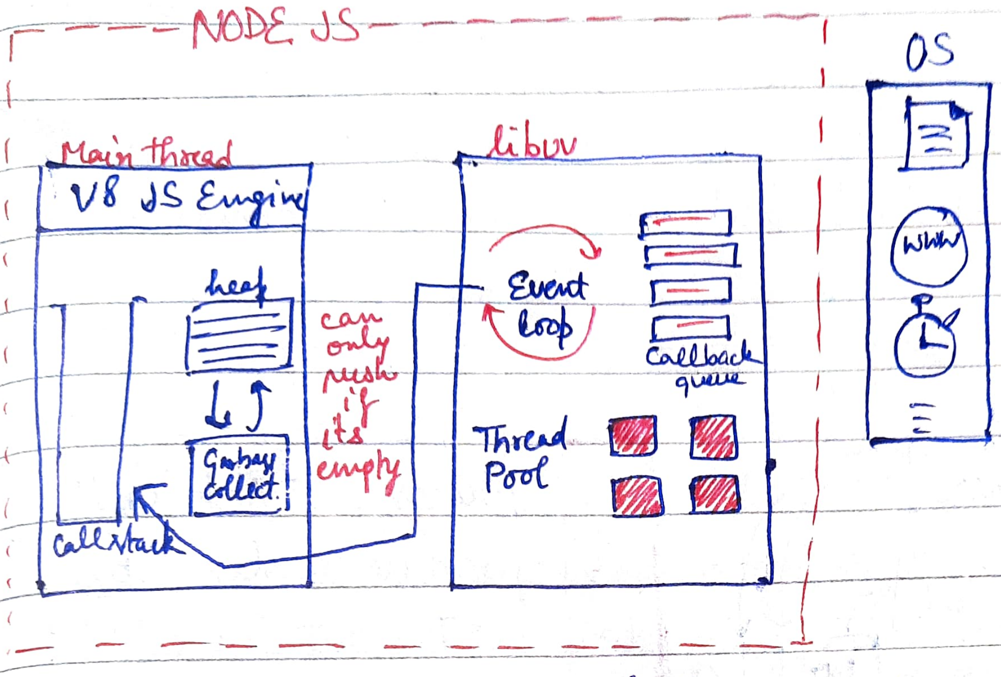
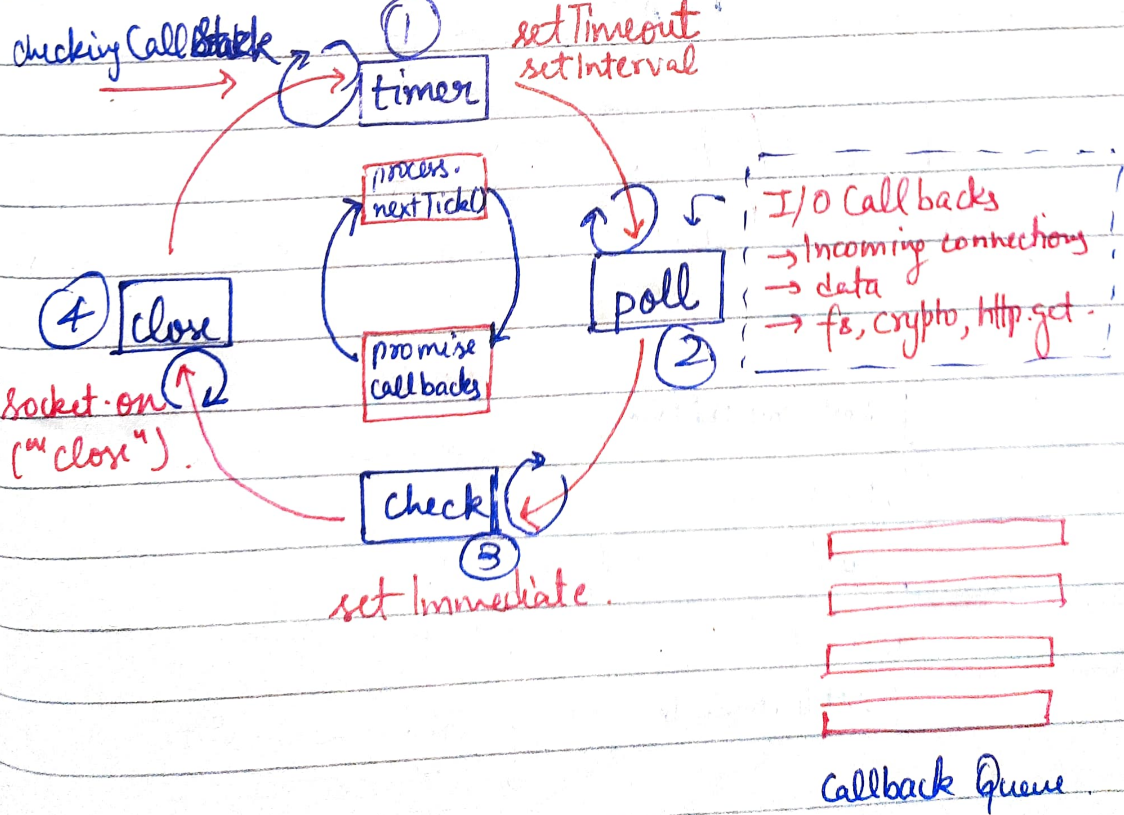
EP-9 LIBUV & EVENT LOOP.

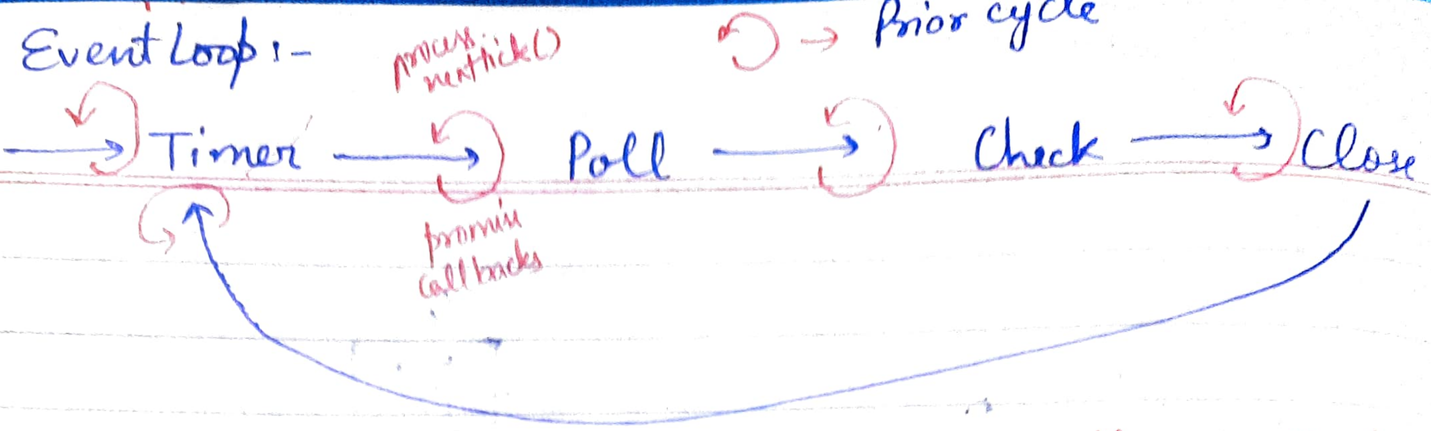


* Event loop continuously checks callback queue & Callstack & whenever call stack is empty it pushes some code to Callstack.

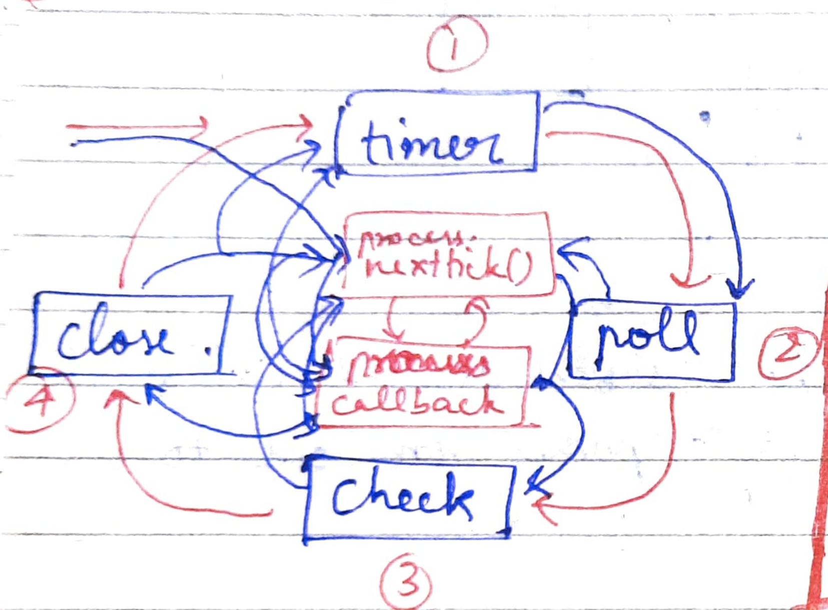
EVENT LOOP



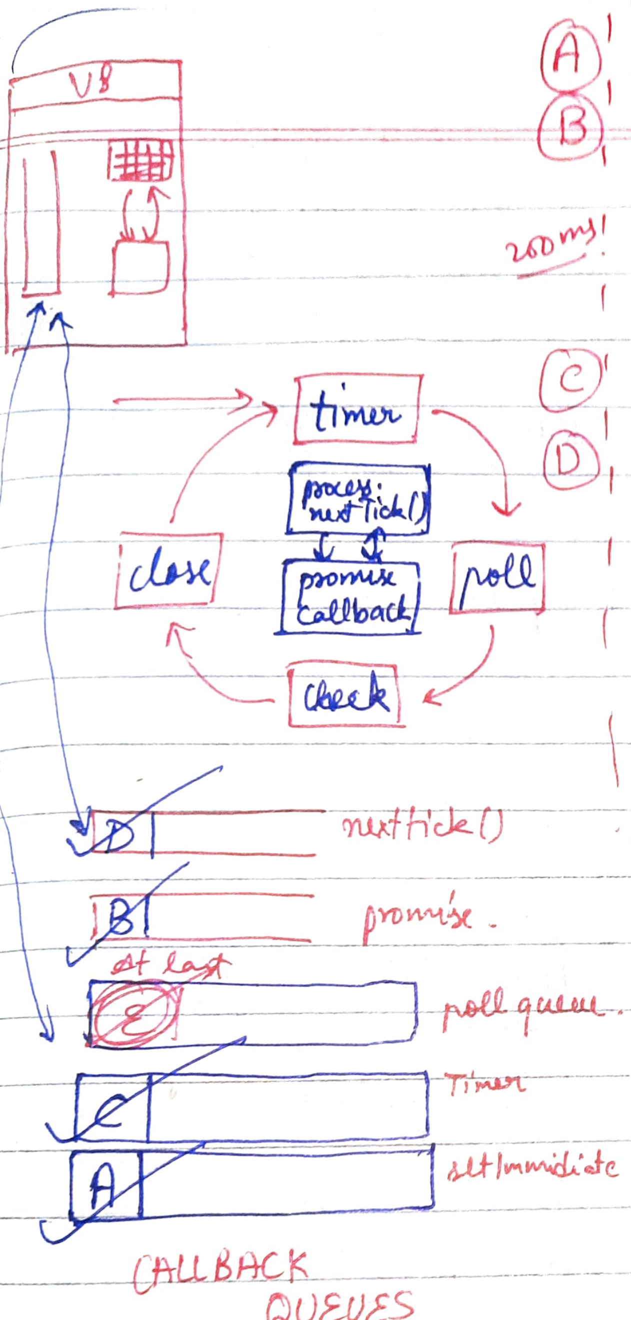
Event Loop :-



* At every step it will check the callbacks if something's there.



1	process.nextTick(cb);
2	promise.resolve (cb);
3	setTimeout (cb, 0);
5	setImmidiate(cb);
4	fs.fileread (" ", cb);
4	https.get ("URL", cb);



CALLBACK QUEUES

```
    const a = 100;
(A) setImmidiate(() => clg());
(B) Promise.resolve(() => clg());
    fs.readfile(" ", "", () => {
200ms!        clg("file");
    });
(C) setTimeout(() => clg("Timer"), 0);
(D) process.nextTick(() => clg("proce"));
    function printA() {
          clg(" a=", a);
    }
    printA();
    clg(" Last ");
```

CONSOLE

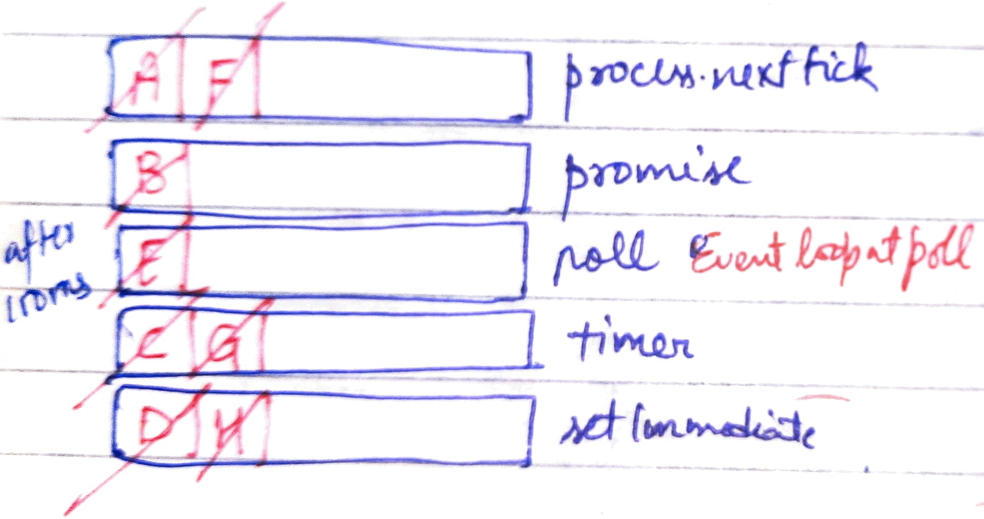
```
> a = 100
. Last line
. ~~timer expired~~
. process.nextTick
. promise
. timer expired
  set Immidiate
. file read
```

EVENT LOOP WAITS AT POLL PHASE IF ALL THE QUEUES ARE EMPTY

SemiInfinite Loop, when it has nothing to do it waits at poll phase. (When UV_queues are empty)

LOOP STARTS FROM POL

SO, EXECUTION ALSO FROM POLL



console

>last line of file

next tick

promise

Timer expired.

setImmediate.

File reading CB.

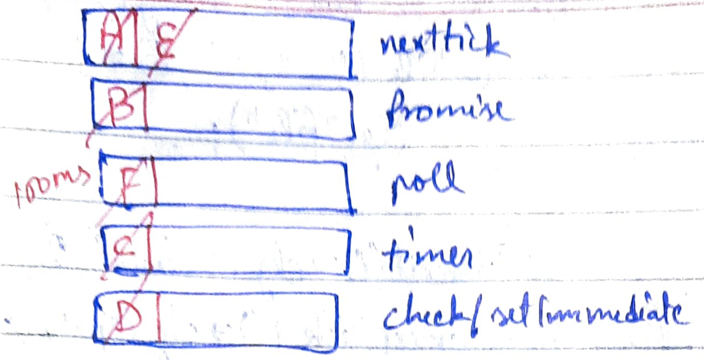
2nd nextTick.

~~2nd timer~~

2nd setImmidiate.

2nd timer

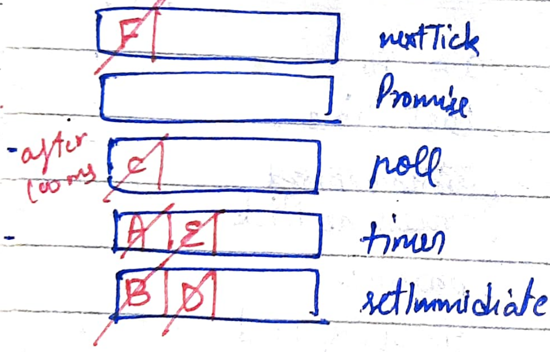
~~nested NextTick~~



> Last line of file
nexttick
innernexttick
promise
timer
setImmediate
file reading(fs)

NESTED NEXTICK WILL BE EXECUT FIRST

Ex:



Last line of file
a = 100;
Timer Expired.
next Tick
set Immidiate.
2nd " "
2nd Timeout.
file read

NOTE:

* One tick is the one cycle of the Event Loop.

There are 2 more phases b/w timers & poll.

pending callbacks → executes I/O callbacks deffered to next loop iteration

idle, prepare → only used internally
↳ some checks for poll phase